Indian Institute of Management, Jammu

Business Analytics

Topic: Predicting Customer Churn Rates for OTT Platforms

PROJECT SUBMITTED TO - Prof. Sundar
Group No: 8
Members: **MBA24026 - Akash Gupta** **MBA24123 - Kayala Siva Ram Prakhash** **MBA24166 - Neha Ajay Maktedar** **MBA24217 - Sahar Guleria** **MBA24253 - Shrishti Raizada** **MBA24298 - Vikansha** **MBA24305 - Yash Joshi**

Abstract

The exponential growth of Over-The-Top (OTT) platforms has led to increased competition in retaining customers, and therefore, customer churn prediction is the most critical focus for profitability and sustainability. This project explores predictive models to identify churn tendencies among OTT platform users, using a dataset of 2,000 instances with features such as viewing habits, engagement metrics, and customer support interactions. These included the use of several machine learning algorithms, which are Gradient Boosting Machine (GBM), Random Forest, Decision Tree, Logistic Regression, Naive Bayes, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Neural Networks to predict churn by taking into consideration challenges like class imbalance and irrelevant features.

The best-performing models are Random Forest and Gradient Boosting, which have got a balanced trade-off across performance metrics: accuracy, precision, recall, and F1-score. Class imbalance can be handled using the SMOTE technique, and feature engineering shows weekly minutes watched and customer support calls to be critical predictors. Model strengths notwithstanding, limited recall for churn prediction needs optimization via hyperparameter tuning as well as advanced resampling techniques.

The results highlight the potential of predictive analytics in transforming churn management from reactive to proactive, equipping OTT platforms with actionable insights to enhance retention strategies and customer satisfaction. Recommendations include deploying ensemble models, improving dataset balance, and optimizing features for better engagement and dissatisfaction monitoring. This project offers a robust framework for predictive modeling in a competitive OTT market, ensuring long-term business success.

Introduction

The competition to retain customers has increased with the exponential growth of Over-The-Top (OTT) platforms. Customer churn, or when subscribers discontinue their service, is a critical issue for OTT providers in pursuit of long-term sustainability and profitability. The accurate prediction of customer churn enables businesses to adopt targeted strategies to retain their customer base, thus reducing losses and maximizing revenue.

This project focuses on predictive models to identify churn tendencies among OTT platform users. We have utilized a dataset of key predictors that include weekly minutes watched, maximum daily minutes, and customer support calls to examine and compare the different regression and classification techniques. We will use the Gradient Boosting Machine (GBM), Decision Tree, Random Forest, Logistic Regression, Naive Bayes, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and neural networks. Finally, the best-performing model must be determined based on its prediction accuracy of churn while overcoming difficulties such as unbalanced data and irrelevant features.

Our study would not only identify the contributing factors of churn but also indicate the practical interventions that OTT platforms can perform to enhance retention. Predictive analytics will move businesses from the reactive approach towards proactive customer interaction, which, in turn, will help achieve long-term success in a saturated market.

Problem Statement

Predicting Customer Churn Rates for OTT Platforms

The OTT market is perfectly competitive and the retention of customers is a must for the companies to stay afloat in the long run. This project is about building a model to predict which subscribers are likely going to churn, based on viewing habits, engagement metrics and demographic information. OTT platforms can take preventive measures to retain customers by identifying at-risk customers and suggesting measures for reducing churn that increases customer lifetime value and revenue stability.

Motivation for solving the problem

It costs 3 to 10 times more for a business to acquire a new subscriber then to keep their existing fans. The insight of why and how subscribers churn will help OTT platforms define better retention strategies, personalize the content by viewing patterns and enhance customer satisfaction. It will assist in driving targeted retention, resource allocation, and a robust competitive position.

Data Collection Overview

The dataset used for this analysis was sourced from **Kaggle**, a well-known platform for datasets, competitions, and collaborative data science projects. Kaggle datasets are typically curated by experts or sourced from real-world business problems, ensuring a level of reliability and practical relevance.

The primary objective of the dataset is to identify customer churn, defined as the phenomenon where customers discontinue using a company's product or service. Understanding churn is critical for businesses to implement retention strategies and improve customer lifetime value.

Dataset Characteristics:

1. **Sample Size**:
   The dataset comprises **2,000 instances**, split into two classes:
   - **Non-Churned Customers (1,738 instances)**: Representing customers who remained with the service.
   - **Churned Customers (262 instances)**: Representing customers who discontinued their service.
2. **Class Imbalance**:
   There is a clear class imbalance, with non-churned customers making up approximately 87% of the data and churned customers accounting for 13%. This imbalance presents challenges for predictive modeling, as standard machine learning algorithms tend to favor the majority class, leading to suboptimal performance on the minority class.
3. **Features and Attributes**:
   The dataset includes multiple features, such as demographic details, usage patterns, and customer support interactions. These features provide valuable insights into the predictors of churn.
   Examples of features might include:
   - **Usage metrics**: Total minutes watched, maximum daily minutes, minimum daily minutes.
   - **Customer support interactions**: Number of calls made to customer service.
   - **Engagement metrics**: Videos watched, multi-screen usage.
4. **Source Credibility**:
   Kaggle datasets often include metadata, descriptions, and discussions from contributors, which validate their authenticity. Researchers and businesses frequently rely on these datasets for exploratory analysis, benchmarking, and model development.

Methodology

Preprocessing:

Missing Data Handling: Missing values were imputed to ensure completeness of data.
Feature Scaling and Encoding: Continuous features were normalized, and categorical features were encoded to optimize model performance.

Class Imbalance Handling:

Techniques such as Synthetic Minority Oversampling (SMOTE) were considered to improve the representation of the minority class (churned customers).

Model Selection and Training:

All of the above models were used: Logistic Regression, Naive Bayes, Decision Tree, K-Nearest Neighbors (KNN), Gradient Boosting Machine (GBM), Random Forest, and Support Vector Machine (SVM), Neural Networks were used.
Further, all the models accuracy were compared using Random Forest stacking and conclusions were drawn.
Hyperparameter optimization was performed to tune the models to increase precision, recall, and other performance metrics.

Evaluation Metrics:

Accuracy, Precision, Recall, F1 Score, and Geometric Mean were the metrics used for model evaluation.
Special emphasis was given to metrics such as Recall and F1 Score for the churn class to reduce the impact of the imbalanced dataset.
Insights Through Confusion Matrix:

Confusion matrices gave detailed insights into true positives, true negatives, false positives, and false negatives for each model, giving clarity on areas that needed improvement.
Feature Importance Analysis:

Key predictors such as weekly minutes watched, maximum daily minutes, and customer support calls were found to be the most influential drivers of churn.

Post-Modeling Analysis

Strengths and Weaknesses of Models:

Random Forest and Gradient Boosting Machine were the best models for dealing with imbalanced data, balancing precision and recall.
Logistic Regression and Naive Bayes were not good at dealing with complex patterns and imbalanced datasets.

Optimization Recommendations:

Employing ensemble techniques, resampling, and hyperparameter tuning to further improve performance.
Using advanced models such as Neural Networks for larger datasets with higher computational resources.

Source :
https://www.kaggle.com/datasets/dvijkalsi/customer-churn-ott?resource=download&select=customer_data.csv

Experiment and results - Techniques used, results, and Discussion

ANOVA

	df	SS	MS	F	Significance F
Regression	12	33.5712583	2.797604855	28.6380617	8.69051E-61
Residual	1987	194.106742	0.097688345		
Total	1999	227.678			

	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%	Lower 95.0%	Upper 95.0%
Intercept	-0.18116377	0.06145178	-2.94806391	0.003235	-0.30168045	-0.06064709	-0.30168045	-0.06064709
gender	0.015174935	0.01407248	1.078341041	0.28101248	-0.01242343	0.042773303	-0.01242343	0.042773303
age	0.0002885	0.00068636	0.420333275	0.67428746	-0.00105756	0.001634561	-0.00105756	0.001634561
0_of_days_subscribed	-9.2048E-05	0.00017621	-0.52236911	0.60147159	-0.00043763	0.000253534	-0.00043763	0.000253534
multi_screen	0.28103169	0.02353144	11.9428189	8.4344E-32	0.23488281	0.327180569	0.23488281	0.327180569
mail_subscribed	-0.06656206	0.01552061	-4.28862487	1.884E-05	-0.09700044	-0.03612369	-0.09700044	-0.03612369
weekly_mins_watched	-0.00409378	0.27916039	-0.01466463	0.98830121	-0.55157158	0.543384012	-0.55157158	0.543384012
minimum_daily_mins	0.01810435	0.00658132	2.750870305	0.00599763	0.005197343	0.031011356	0.005197343	0.031011356
maximum_daily_mins	0.041983853	2.46318585	0.017044533	0.9864028	-4.78871425	4.872681957	-4.78871425	4.872681957
weekly_max_night_mins	-7.3025E-05	0.00035908	-0.20336824	0.83886805	-0.00077724	0.000631188	-0.00077724	0.000631188
videos_watched	-0.00447787	0.0028186	-1.58868879	0.11228976	-0.01000558	0.001049842	-0.01000558	0.001049842
maximum_days_inactive	-0.03911958	0.02278614	-1.7168148	0.08616886	-0.08380682	0.005567652	-0.08380682	0.005567652
customer_support_calls	0.054678371	0.00532102	10.27591876	3.5995E-24	0.044243007	0.065113736	0.044243007	0.065113736

Significant variables on the basis of p-value (< 0.05)

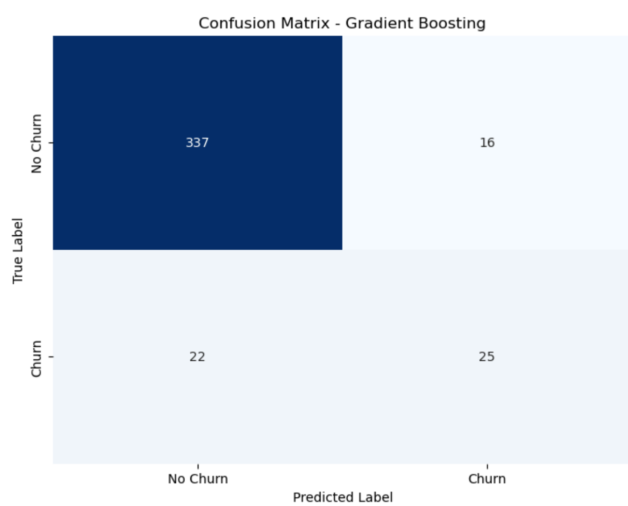
We ran a regression on the dataset to generate a p-value for all the variables being used to predict the churn rate.
We observed that multi_screen, mail_subscribed, minimum_daily_mins, and customer_support_calls came out to be significant variables.

Gradient Boosting Machine

Key Insights:

- **Accuracy**: This model has demonstrated an overall **accuracy** of **91%,** which demonstrates that the model can predict the churn or non-churn rate correctly for 91% of test data. However, in case of an imbalanced dataset accuracy will not provide a complete picture hence throwing light on precision and recall is also important.
- **Precision and Recall**:
  - **Precision (0.61)** for churn predictions (class 1) is relatively low, thus depicting that 61% of customers predicted churned were actually churned. Hence the number of **false positives** is comparatively high.
  - **Recall (0.53)** for churn predictions is still better, which implies that the model was able to identify 53% of actual churn cases correctly. This shows that nearly half of the churned customers were missed (i.e. **false negatives**).
- **F1 Score**: The **F1 Score (0.57)**, which balances precision and recall, depicts that this model is struggling when it comes to predicting the churn rate especially when it comes to identifying the churned customers.

- **Confusion Matrix**:



  - 337 cases were correctly identified from a set of 353 non-churn cases while 16 cases were wrongly identified as churn.
  - 25 out of 47 cases were correctly classified as churned meanwhile 22 cases were missed highlighting the issue of lower recall value when it comes to churn predictions.

Conclusion:

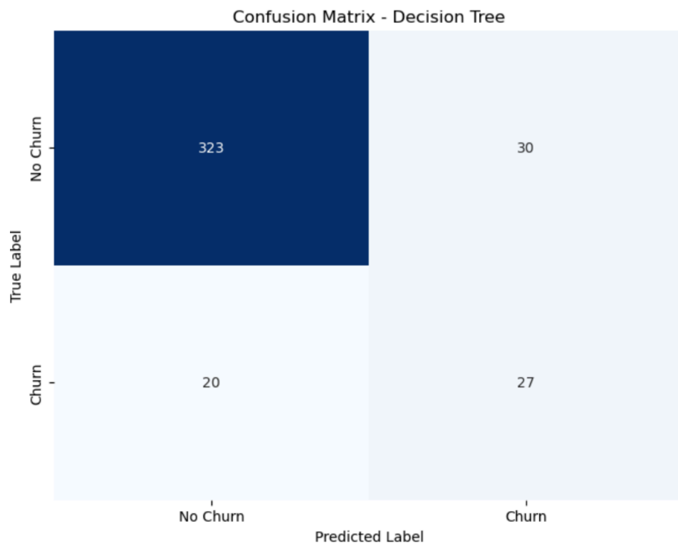
We can conclude from our observations that the model performs well when it comes to predicting the non-churned cases in comparison to churned cases. **Class imbalance** can be one of the main reasons since the number of non-churned cases is significantly higher when compared to churned cases hence techniques like **hyperparameter tuning** to increase the values of recall and precision for churned customers.

Decision Tree

Key Insights:

- **Accuracy**: This model has demonstrated an overall **accuracy** of **88%,** which demonstrates that the model can predict the churn or non-churn rate correctly for 88% of test data. However in case of imbalanced dataset accuracy will not provide a complete picture hence throwing light on precision and recall is also important.
- **Precision and Recall**:
    - **Precision (0.47)** for churn predictions (class 1) is relatively very low, thus depicting that 47% of customers predicted churned were actually churned. Hence the number of **false positives** is comparatively high.
    - **Recall (0.57)** for churn predictions is low, which implies that the model was able to identify only 53% of actual churn cases correctly. This shows that almost half of the churned customers were missed (i.e. **false negatives**).
- **F1 Score**: The **F1 Score (0.57)**, which balances precision and recall, depicts that this model is struggling when it comes to predicting the churn rate.

- **Confusion Matrix**:



- ○ 323 cases were correctly identified from a set of 353 non-churn cases while 30 cases were wrongly identified as churn.
- ○ 27 out of 47 cases were correctly classified as churned meanwhile 20 cases were missed highlighting the issue of lower recall value when it comes to churn predictions.

Conclusion:

We can conclude from our observations that the model performs well when it comes to predicting the non-churned cases in comparison to churned cases. **Class imbalance** can be one of the main reasons since the number of non-churned cases is significantly higher when compared to churned cases hence oversampling techniques like **SMOTE** or adjusting decision trees depth can be done.

Logistic Regression

Key Insights:

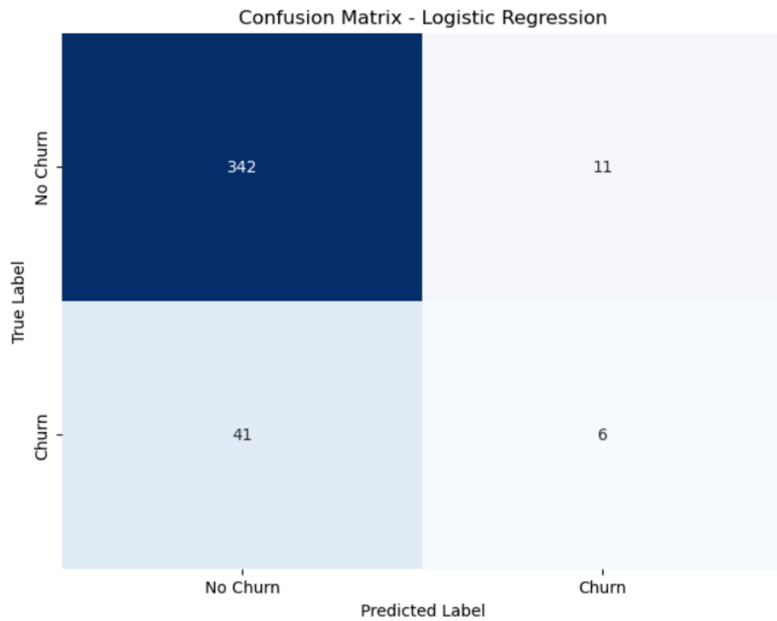
- **Accuracy**: This model has demonstrated an overall **accuracy** of **87%,** which demonstrates that the model can predict the churn or non-churn rate correctly for 88% of test data. However in case of imbalanced dataset accuracy will not provide a complete picture hence throwing light on precision and recall is also important. Moreover by judging it on more than one classification metrics we can conclude that the minority class which is churn in this case suffers.
- **Precision and Recall**:

    - **Precision (0.35)** for churn predictions (class 1) is very low, thus depicting that 35% of customers predicted churned were actually churned. Hence the number of **false positives** is comparatively high.
    - **Recall (0.13)** for churn predictions is very low, which implies that the model was able to identify only 13% of actual churn cases correctly. This shows that almost all of the churned customers were missed (i.e. **false negatives**).
- **F1 Score**: The **F1 Score (0.19)**, which balances precision and recall, depicting its ability to predict the churn rate also highlighting the observable bias that this model has towards non-churn cases.
- **Odds Ratio:** The odds ratio analysis gives critical insights into the factors that influence churn among customers. In particular, multi-screen usage increases the probability of churn considerably, with an odds ratio of 6.26, meaning customers who use more than one screen are six times more likely to churn. Repeated calls for customer support are also associated with higher churn rates, which reflects underlying dissatisfaction. On the other hand, features such as mail subscriptions and inactive days reduced have strong mitigation effects on churn with odds ratios of 0.49 and 0.56, respectively. The results suggest that multi-screen users should be targeted with customized retention strategies and mail subscriptions and engagement initiatives should be utilized to boost customer loyalty.

Odds Ratios:

	Feature	Coefficient	Odds Ratio
4	multi_screen	1.834845	6.264161
12	customer_support_calls	0.489678	1.631790
7	minimum_daily_mins	0.240278	1.271602
6	weekly_mins_watched	0.008875	1.008914
9	weekly_max_night_mins	0.001502	1.001503
3	no_of_days_subscribed	0.000430	1.000430
2	age	-0.000099	0.999901
8	maximum_daily_mins	-0.001379	0.998622
0	year	-0.002904	0.997100
10	videos_watched	-0.049083	0.952102
1	gender	-0.206349	0.813549
11	maximum_days_inactive	-0.575782	0.562265
5	mail_subscribed	-0.717786	0.487831

- **Confusion Matrix**:



- ○ 342 cases were correctly identified from a set of 353 non-churn cases while 11 cases were wrongly identified as churn.
- ○ 6 out of 47 cases were correctly classified as churned meanwhile 41 cases were missed highlighting the issue of extremely low recall value when it comes to churn predictions.

Conclusion:

We can conclude from our observations that the model performs well when it comes to predicting the non-churned cases in comparison to churned cases. **Class imbalance** can be one of the main reasons since the number of non-churned cases is significantly higher when compared to churned cases hence **class imbalance handling**, **feature engineering techniques** could be applied or better models like **Random Forest** or **Gradient Boosting Machine** can be used. Model's decision threshold can be adjusted to improve the recall values.

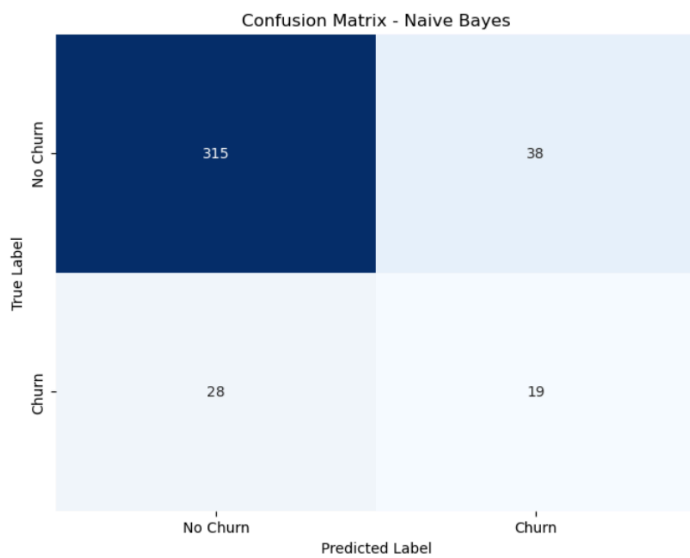
Naive Bayes

Key Insights:

- **Accuracy**: This model has demonstrated an overall **accuracy** of **83%,** which demonstrates that the model can predict the churn or non-churn rate correctly for 83% of test data. However in case of imbalanced dataset accuracy will not provide a complete picture hence throwing light on precision and recall is also important. Moreover by judging it on more than one classification metrics we can conclude that the minority class which is churn is affected.

- **Precision and Recall**:
  - **Precision (0.33)** for churn predictions (class 1) is very low, thus depicting that 33% of customers predicted churned were actually churned. Hence the number of **false positives** is comparatively high.
  - **Recall (0.40)** for churn predictions is low, which implies that the model was able to identify 40% of actual churn cases correctly. This shows that the majority of the churned customers (60%) were missed (i.e. **false negatives**).
- **F1 Score**: The **F1 Score (0.37)**, which balances precision and recall, depicting that the model is not very effective in predicting the churn rate.

- **Confusion Matrix**:



  - 315 cases were correctly identified from a set of 353 non-churn cases while 38 cases were wrongly identified as churn.
  - 19 out of 47 cases were correctly classified as churned thus confirming the weakness of the minority class.

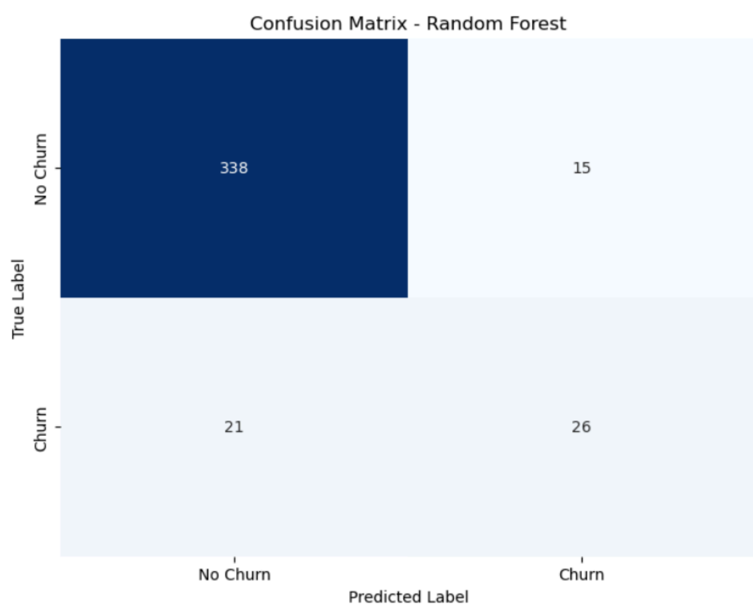
Conclusion:

We can conclude from our observations that the model performs well when it comes to predicting the non-churned cases in comparison to churned cases. **Class imbalance** can be one of the main reasons since the number of non-churned cases is significantly higher when compared to churned cases hence **class imbalance handling**, **hyperparameter tuning** could be applied or better models like **Random Forest** or **Gradient Boosting Machine** could be explored.

Random Forest

Key Insights:

- **Accuracy**: This model has demonstrated an overall **accuracy** of **91%,** which demonstrates that the model can predict the churn or non-churn rate correctly for 91% of test data. However in case of imbalanced dataset accuracy will not provide a complete picture hence throwing light on precision and recall is also important.
- **Precision and Recall**:
  - **Precision (0.63)** for churn predictions (class 1) is relatively low, thus depicting that 63% of customers predicted churned were actually churned. Hence the number of **false positives** is high.
  - **Recall (0.55)** for churn predictions is still better, which implies that the model was able to identify 55% of actual churn cases correctly. This shows that nearly half of the churned customers were missed (i.e. **false negatives**).
- **F1 Score**: The **F1 Score (0.57)**, which indicates that the model strikes a good balance between precision and recall for churn predictions, but there is still a room for improvement when it comes to recall.
- **Confusion Matrix**:



  - 338 cases were correctly identified from a set of 353 non-churn cases while 15 cases were wrongly identified as churn.
  - 26 out of 47 cases were correctly classified as churned meanwhile 21 cases were missed highlighting the issue of lower recall value when it comes to churn predictions.

Conclusion:

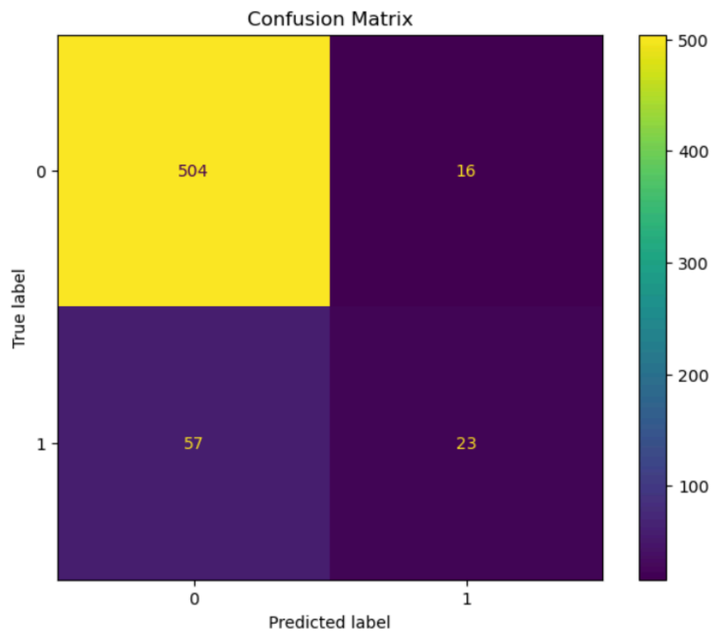
We can conclude from our observations that the Random Forest model performs well when it comes to predicting the non-churned cases in comparison to churned cases. **Class imbalance** can be one of the main reasons since the number of non-churned cases is significantly higher when compared to churned cases. Techniques such as **addressing class imbalance** (e.g., using **SMOTE** for oversampling the minority class) or adjusting **hyperparameters** could be considered for improving model's predictability. Additionally, exploring **other advanced models** like **Gradient Boosting** or **Neural Networks** might provide better results.

K-Nearest Neighbours

Key Insights:

- **Accuracy:** This model has demonstrated an overall **accuracy** of **87%,** which demonstrates that the model can predict the churn or non-churn rate correctly for 87% of test data. However in case of imbalanced dataset accuracy will not provide a complete picture hence throwing light on precision and recall is also important.
- **Precision and Recall**:
  - **Precision (0.59)** for churn predictions (class 1) is relatively low, thus depicting that 59% of customers predicted churned were actually churned. Hence the number of **false positives** is high.
  - **Recall (0.29)** for churn predictions is very low, which implies that the model was able to identify only 29% of actual churn cases correctly. This shows that almost all of the churned customers were missed (i.e. **false negatives**).
- **F1 Score**: The **F1 Score (0.39)**, which balances precision and recall, depicting that the model is not very effective in predicting the churn rate.
- **Geometric Mean (0.53)** shows a moderate balance between correctly identifying both classes 0 and 1. Poor recall contributes to affecting the model's generalizability when it comes to minority class.

- **Confusion Matrix**



True Positives (TP): 23 - which means 23 cases of churn were correctly predicted.

True Negatives (TN): 504 - which means 504 cases of non-churn were correctly predicted.

False Positives (FP): 16 - which means 16 cases of churn were wrongly predicted.

False Negatives (FN): 57 - which means 57 cases of non-churn were wrongly predicted.

This highlights the strong biases towards the majority class (non-churn).

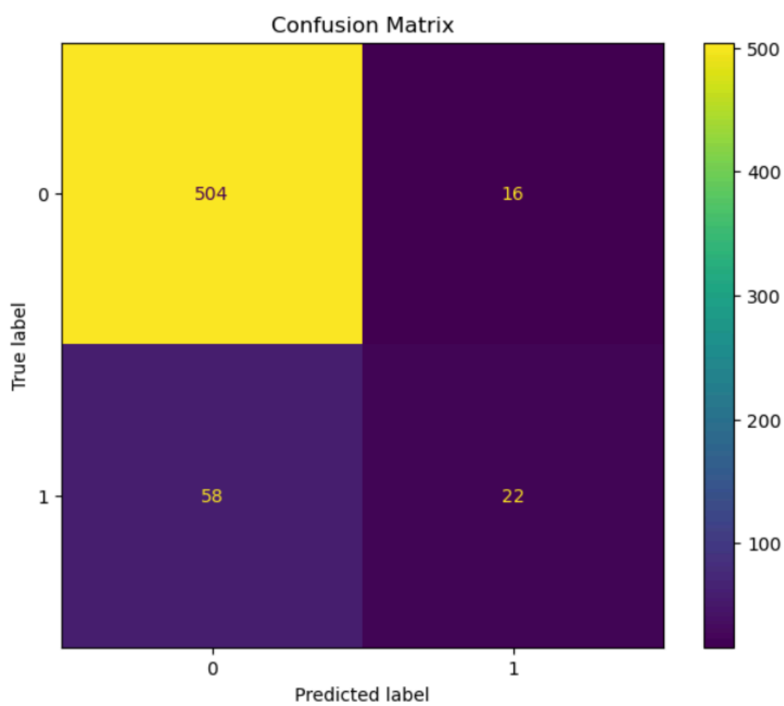
Conclusion :

We can conclude from our observations that the KNN model performs well when it comes to predicting the non-churned cases in comparison to churned cases. **Class imbalance** can be one of the main reasons since the number of non-churned cases is significantly higher when compared to churned cases. Techniques such as **addressing class imbalance** (e.g., using **SMOTE** for oversampling the minority class), **feature engineering** or o**ptimizing KNN parameters** could be considered for improving model's predictability. Additionally, exploring **other advanced models** like **Gradient Boosting** or **Neural Networks** might provide better results.

Support Vector Machine

Key Insights:

- **Accuracy:** This model has demonstrated an overall **accuracy** of **88%,** which demonstrates that the model can predict the churn or non-churn rate correctly for 88% of test data. However in case of imbalanced dataset accuracy will not provide a complete picture hence throwing light on precision and recall is also important.
- **Precision and Recall**:
    - **Precision (0.58)** for churn predictions (class 1) is relatively low, thus depicting that 58% of customers predicted churned were actually churned. Hence the number of **false positives** is high.
    - **Recall (0.28)** for churn predictions is very low, which implies that the model was able to identify only 28% of actual churn cases correctly. This shows that almost all of the churned customers were missed (i.e. **false negatives**).
- **F1 Score**: The **F1 Score (0.39)**, which balances precision and recall, depicts that the model is not very effective in predicting the churn rate and has a poor performance on minority class.
- **Geometric Mean (0.52)** shows a moderate balance between correctly identifying both classes 0 and 1. Poor recall contributes to affecting the model's generalizability when it comes to minority class.
- **Confusion Matrix**



True Positives (TP): 22 - which means 22 cases of churn were correctly predicted.

True Negatives (TN): 504 - which means 504 cases of non-churn were correctly predicted.

False Positives (FP): 16 - which means 16 cases of churn were wrongly predicted.

False Negatives (FN): 58 - which means 58 cases of non-churn were wrongly predicted.

This highlights the strong biases towards the majority class (non-churn).

Conclusion :

We can conclude from our observations that the SVM model performs well when it comes to predicting the non-churned cases in comparison to churned cases. **Class imbalance** can be one of the main reasons since the number of non-churned cases is significantly higher when compared to churned cases. Techniques such as **addressing class imbalance** (e.g., using **SMOTE** for oversampling the minority class), **feature engineering** or o**ptimizing KNN parameters** could be considered for improving model's predictability. Evaluating the model using metrics like **PR AUC** ensures a more comprehensive assessment.

Neural Networks Classifier

Hidden Neurons	Batch Size	Accuracy	F1 Score	G-Mean	TP	FP	FN	TN
16	16	0.888	0.511	0.647	35	22	45	498
16	32	0.868	0.415	0.576	28	27	52	493
16	64	0.863	0.359	0.523	23	25	57	495
32	16	0.880	0.493	0.644	35	27	45	493
32	32	0.882	0.496	0.645	35	26	45	494
32	64	0.875	0.436	0.588	29	24	51	496

```
print(results_df)
```

```
   hidden_neurons  batch_size  accuracy  f1_score  geometric_mean  TP  FP  FN  \
0              16          16  0.876667  0.421875        0.569096  27  21  53
1              16          32  0.880000  0.470588        0.617688  32  24  48
2              16          64  0.878333  0.406504        0.549256  25  18  55
3              32          16  0.881667  0.481752        0.627265  33  24  47
4              32          32  0.881667  0.474074        0.618310  32  23  48
5              32          64  0.876667  0.430769        0.578958  28  22  52

    TN
0  499
1  496
2  502
3  496
4  497
5  498
```

1. Accuracy

- Accuracy ranges from 0.876 to 0.881, which implies a consistent overall performance.
- However this high accuracy can be misleading since the dataset is imbalanced favouring the majority class.

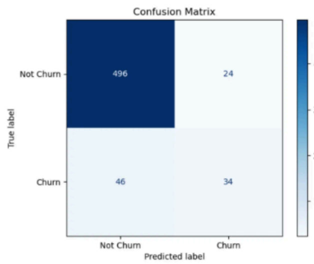
2. F1 Score

- F1 scores range between 0.42 and 0.48, demonstrating the model's inability to strike a balance between precision and recall.
- The highest F1 score (0.481752) is observed for the configuration with 32 hidden neurons and batch size 16, showing a better performance for prediction.

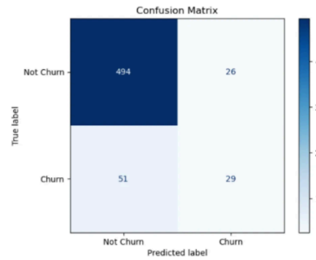
3. Geometric Mean

- The G-Mean values range between 0.54 and 0.62, showing a better balance between recall and specificity.
- The highest G-Mean (0.627265) is achieved with 32 hidden neurons and batch size 16, which gives a better overall performance of the model when compared with others.

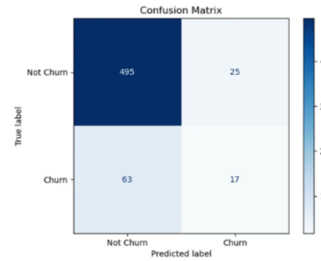
- **Confusion Matrix**



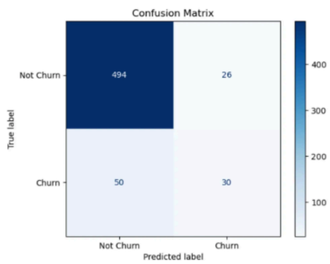
`16 neurons and batch size 16`



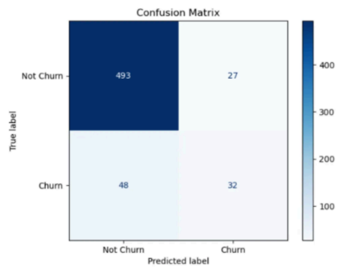
`16 neurons and batch size 32`



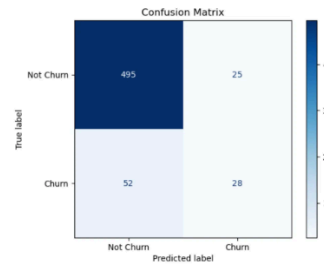
`16 neurons and batch size 64`



`32 neurons and batch size 16`



`32 neurons and batch size 32`



`32 neurons and batch size 64`

True Positives (TP): [**25 and 33]**, indicating the number of correctly identified churners. The highest TP count (**33**) occurs with **32 hidden neurons and batch size 16**.

True Negatives (TN): [**47 and 55]**, representing missed churners. The lowest count (**47**) also corresponds to **32 hidden neurons and batch size 16**, showing a better recall.

False Positives (FP): [**21 and 24]**, indicating misclassifications of non-churners as churners.

False Negatives (FN): [**496 and 502]**, reflecting strong performance when it comes to identifying the majority class (non-churners). This highlights the strong biases towards the majority class (non-churn).

Conclusion :

The configuration of 32 hidden neurons with a batch size of 16 has an overall best performance. The F1-score and G-Mean of the same is 0.4817 and 0.6272 respectively suggesting that the model strikes a good balance between precision and recall. Techniques such as **addressing class imbalance** (e.g., using **SMOTE** for oversampling the minority class), **feature engineering** or **optimizing KNN parameters** could be considered for improving model's predictability. Evaluating the model using metrics like **PR AUC** ensures a more comprehensive assessment. **Hyperparameter tuning** (e.g., neurons, batch size, learning rate) using grid or random search can enhance results. **Ensemble methods like Bagging, Boosting** (e.g., XGBoost, AdaBoost), or Stacking could be incorporated to handle class imbalance.

Stacking

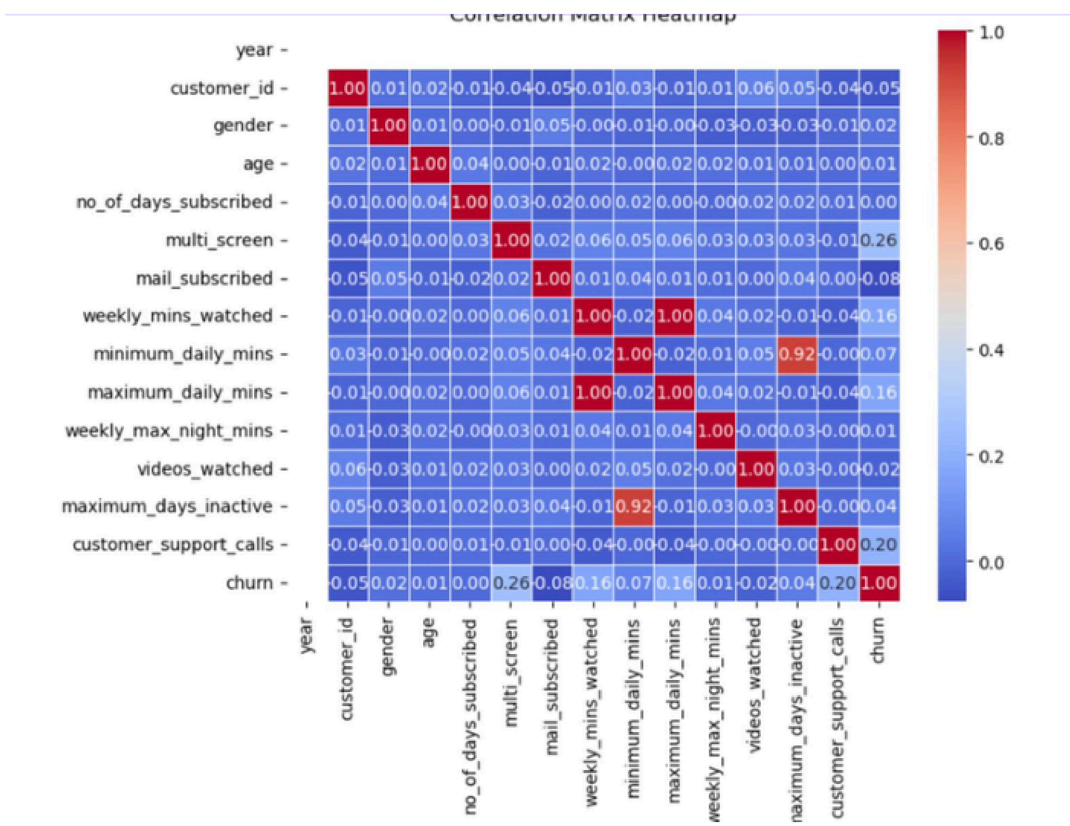
Stacking is an ensemble approach that tries to improve the overall predictive performance by combining the prediction results of multiple models. We have used this method in our project to do a comparative analysis of all the classifiers used.

Stacking Implementation:

```
Classifier Performance (Accuracy):
Random Forest: 91.00%
Gradient Boosting: 90.50%
SVM: 88.00%
KNN: 88.00%
Neural Network: 87.75%
Decision Tree: 87.75%
Logistic Regression: 87.00%
Naive Bayes: 83.50%
```

- The classifiers we used are selected as base models for implementing the stacking method of ensemble approach.
- The final predictions were made after the outputs of the base model were trained on the meta model.

Correlation Matrix



From the correlation matrix we conclude that Naive Bayes had a poor performance amongst all classifiers since the features are strongly correlated with each other.

- maximum_daily_mins and weekly_mins_watched are highly correlated (1.0)
- minimum_daily_mins and maximum_days_inactive show a strong positive correlation (0.923740)
- multi_screen (0.258324); weekly_mins_watched (0.162876); maximum_daily_mins (0.162874); customer_support_calls (0.204774) has a modest positive correlation with the target feature churn

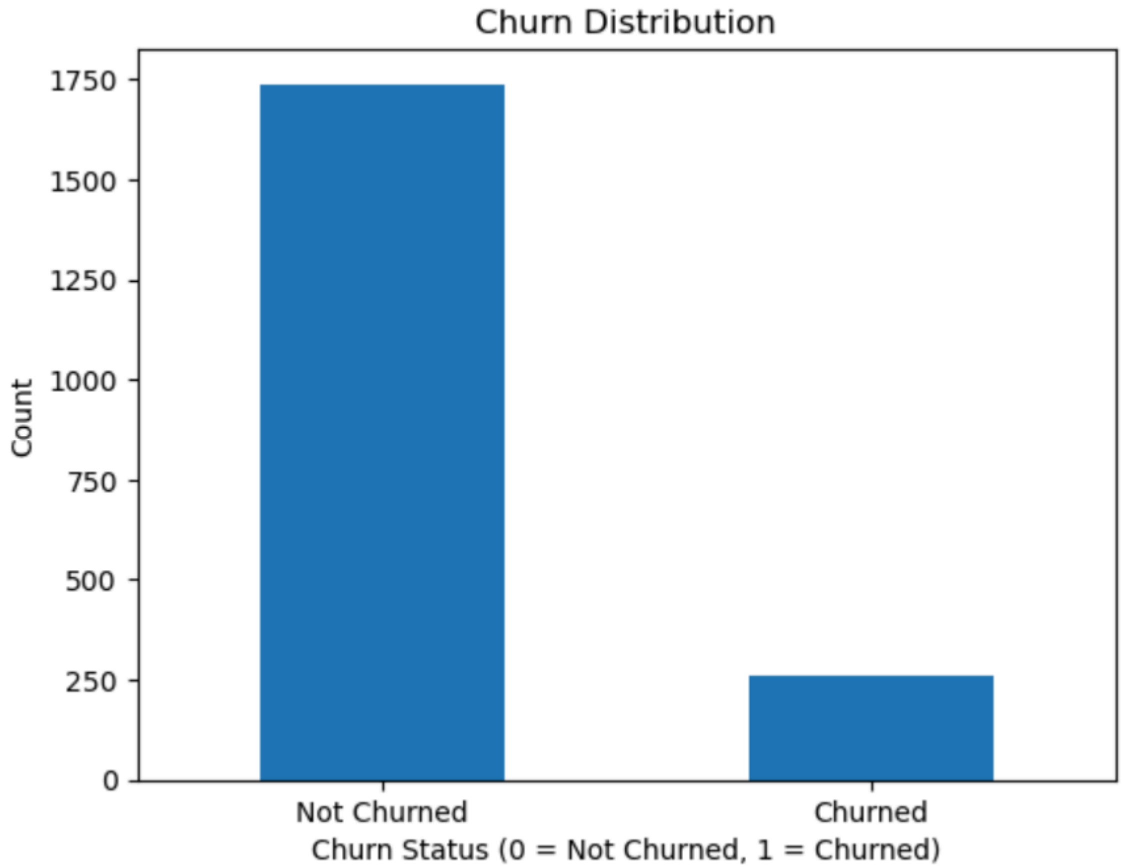
Results and Analysis

- With highest accuracies amongst all the base models and other balanced metrics Random Forest and Gradient Boosting Machine emerged as the strongest models for predicting churn rate.
- Precision value of Logistic Regression was great however it lacked when it came to dealing with recall or identifying the minority class (churn rate).
- SVM and KNN showed moderate performance however had lower recall values due to class imbalance.
- The stacking classifier outperformed most models in both F1 Score and G-Mean by combining the advantages of multiple classifiers.

- We chose Random Forest, Gradient Boosting, and Logistic Regression as base models and Gradient Boosting as the meta-model to give better results.
- This configuration achieved a balanced trade-off between accuracy, precision, recall, and F1 Score.

Final Result

Classifiers	**Gradient Boosting machine**	**Decision Tree**	**Logistic Regression**	**Naïve Bayes**	**Random Forest**	**KNN**	**SVM**	**Neural Networks**
Accuracy	0.91	0.88	0.87	0.83	0.91	0.88	0.88	0.8817
Precision	0.61	0.47	0.35	0.33	0.63	0.59	0.58	0.571
Recall	0.53	0.57	0.13	0.40	0.55	0.29	0.28	0.40
F1-score	0.57	0.52	0.19	0.37	0.59	0.39	0.37	0.470
Geometric Mean	0.73	0.73	0.44	0.62	0.74	0.53	0.52	0.617



Majority Class : Non-Churned Customers (1738)

Minority Class : Churned Customers (262)

Classifiers	**Strengths**	**Weaknesses**	**Implications**
Gradient Boosting machine	High accuracy and geometric mean. Overfitting can be minimized by tuning hyperparameters.	Recall value is moderate suggesting that model is identifying 53% of actual churners.	Overall performance is effective however identifying churners is difficult due to class imbalance.

Decision Tree	Interpretable and has a better recall value when compared with other models.	Precision and geometric mean values are low. Reliability is less due to imbalance in the dataset.	Might predict non-churners as churners. Bagging/Boosting is preferred to improve reliability.
Logistic Regression	This model is simple and it works well in case of linear relationships.	Very low values of precision, recall and F1-score. Not a suitable model for imbalanced datasets.	Not to be used for prediction rate until the dataset is modified through resampling methods or a balanced dataset is used.
Naïve Bayes	Categorical features are handled well. Computationally effective.	Moderate values of all performance metrics. Limited scope when it comes to identifying the underlying patterns.	Good for a baseline model however lacks sophistication needed for high level churn prediction.
Random Forest	High accuracy, precision and geometric mean. Overfitting can be minimized by tuning hyperparameters.	Moderate value of recall and hence can identify more than 50% of churners.	Strong model for predicting churn however modifications must be made in the dataset to improve the recall value.
KNN	Higher accuracy and precision values when compared with other models.	Focuses on majority class hence low recall and geometric mean values. Sensitive to feature scaling.	Limited scope when it comes to an imbalanced dataset like in this case. Can be used for prediction after tuning the data or

		Higher computational cost.	applying feature processing.
SVM	Balanced values of accuracy and precision. Identifies underlying nonlinear patterns within the data in a high dimensional space.	Imbalanced data lead to lower recall and geometric mean values. Sensitive to hyperparameters hence tuning is required.	Limited scope of applications without kernel adjustments.
Neural Networks	High and consistent accuracy. Hyperparameter tuning can optimize recall and other metrics value leading to improvement in prediction.	Model has a high potential of good performance if trained on a large dataset or when optimized through tuning.	Good model for churn prediction if resampling techniques can modify the imbalances in the dataset and enough resources with higher computational power are provided to the classifier.

Key Insights

- Random Forest or Gradient boosting model is best for predicting the churn rate in this case due to the balanced recall, geometric mean and F1-score values.
- These ensemble models can handle imbalanced datasets and can be scaled if the dataset is large or the model complexity is increased. Hyperparameter tuning can give optimized results.
- Resampling techniques such as SMOTE or hyperparameter tuning can be applied on the imbalanced dataset to get better values of key performance metrics.

CONCLUSION

- **Random forest** and **gradient boosting** achieved the highest accuracy of 91%. However, accuracy alone is insufficient to evaluate performance due to the dataset's class imbalance.
- Churn Predictions (Minority Class)
- **Gradient Boosting:** Offers a **moderate recall (53%)** and **precision (61%)** for churn cases resulting in an F1 score of 57%.
- **Random Forest:** Delivers slightly better metrics with a **recall of 55%**, a **precision of 63%** and an F1 score of 59%.
- **SVM and KNN** had a moderate performance given that they had very **low recall values** because of **class imbalance.**
- By featuring all classifiers benefits, stacking manages to defeat most models in terms of F1 Score and G-Mean.

Best Performing Models: **Random Forest and Gradient Boosting** stand out due to their balance of precision recall and F1 score. Both models performed well in terms of the **geometric mean, indicating robust handling of class imbalances.**

RECOMMENDATIONS

- **Model Selection**: Require the Final Model that Predicts Churn to be either Random Forest or Gradient Boosting. Performance of these models is very well in a trade-off between their metrics and scalability
- **Dataset Improvement**:Balance data and improve recall of churn cases using different resampling techniques like SMOTE or undersampling.
- **Model Optimization:** Hyperparameter tuning on Random Forest and Gradient Boosting to capture improvements in their performance especially recall without sacrificing precision should be done.
- **Feature Engineering:** Constraints should be implemented on feature optimization for Engagement (like weekly minutes and maximum daily minutes watched) and Customer Dissatisfaction: customer support calls.